National Research Council Canada
Conseil national de recherches Canada

Division of Electrical Engineering
Division de génie électrique

ERB-1004

Some Results for the Computation of the Modified Bessel Function of the Third Kind

J.B. Campbell

March 1987

Canada

NRCC No. 27482

88-04437/

Copyright 1987 by
National Research Council of Canada

Permission is granted to quote short excerpts and to reproduce figures and tables from this report, provided that the source of such material is fully acknowledged.

Copyright 1987 par
Conseil national de recherches du Canada

Il est permis de citer de courts extraits et de reproduire des figures ou tableaux du présent rapport, à condition d'en identifier clairement la source.

Additional copies are available free of charge from:

Editorial Office, Room 301
Division of Electrical Engineering
National Research Council of Canada
Ottawa, Ontario, Canada
K1A 0R8

Des exemplaires supplémentaires gratuits peuvent être obtenus à l'adresse suivante:

Bureau des publications, Pièce 301
Division de génie électrique
Conseil national de recherches du Canada
Ottawa (Ontario) Canada
K1A 0R8

SOME RESULTS FOR THE COMPUTATION OF THE MODIFIED BESSEL FUNCTION OF THE THIRD KIND

J. B. CAMPBELL

ERB-1004
NRCC No. 27492

MARCH 1987

Table of Contents

Page

Abstract / Résumé v

1. Introduction 1

2. The curve $K_\alpha(x) = M$. 1

3. The curve $e^x K_\beta(x) = M$ 3

4. Numerical results 5

References 6

Subroutine SPHERK 7

Abstract

An efficient method is presented for approximating the order for which overflow occurs in the determination of the Bessel function $K_\mu(x)$ by forward recurrence. We use a linear spline for this purpose. The function $e^x K_\mu(x)$ is treated similarly. A subroutine, SPHERK, for the determination of the Bessel function $K_{M+\frac{1}{2}}(x)$ of real argument is included. M is a non-negative integer.

Résumé

On présente une méthode efficace pour trouver une approximation de l'ordre auquel il y a débordement dans la détermination de la fonction de Bessel $K_\mu(x)$ par récurrence directe. Nous utilisons à cette fin une fonction spline linéaire. La fonction $e^x K_\mu(x)$ est traitée de la même façon. On a inclus un sous-programme, SPHERK, pour la détermination de la fonction de Bessel $K_{M+\frac{1}{2}}(x)$ à argument réel. M est un entier non négatif.

SOME RESULTS FOR THE COMPUTATION OF THE MODIFIED BESSEL FUNCTION OF THE THIRD KIND

J. B. Campbell

1. INTRODUCTION

The Bessel function $K_\mu(x)$ obeys the recurrence relation

[1] $$K_{\eta+1}(x) = \frac{2\eta}{x} K_\eta(x) + K_{\eta-1}(x)$$

The recurrence formula [1] is stable in the forward direction for η and x real and positive, i.e., given values $K_\nu(x)$ and $K_{\nu+1}(x)$ with $\nu \geq -1$, then $K_{\nu+i}$, $i = 2, 3, 4, ...$, can be generated accurately using [1]. Since

$$\lim_{\mu \to \infty} K_\mu(x) = \infty$$

the use of forward recurrence can lead to overflow unless adequate testing is done. If a good lower bound to the value α for which $K_\alpha(x)$ overflows is available, this can be used to produce an efficient algorithm for the forward recurrence since only a few steps will have to be tested for overflow. The subroutine *RBESK* (see refs. 1, 2) which produces a series of functions $e^x K_{\nu+i}(x)$, $i = 0, 1, ..., N$, uses a linear spline function of x as a lower bound to the value of μ for which $e^x K_\mu(x)$ overflows. In this report, this result is discussed and expanded upon.

2. THE CURVE $K_\alpha(x) = M$

The curve $\alpha = \alpha(x)$ defined by the equation

[2] $$K_\alpha(x) = M, \quad M > 0$$

is a concave curve, i.e., $\alpha'(x) > 0$ and $\alpha''(x) < 0$. This can be shown with the aid of the integral representation

[3] $$K_\alpha(x) = \int_0^\infty e^{-x\cosh(t)} \cosh(\alpha t) dt$$

Let

$$f(\alpha, x) = K_\alpha(x) - M$$

Then

[4] $$\alpha' = -\frac{f_x}{f_\alpha} = \frac{\int_0^\infty \cosh(t) e^{-x\cosh(t)} \cosh(\alpha t) dt}{\int_0^\infty t e^{-x\cosh(t)} \sinh(\alpha t) dt} > 0$$

and

$$\alpha'' = -\frac{f_{\alpha\alpha}(\alpha')^2 + 2f_{\alpha x}\alpha' + f_{xx}}{f_\alpha}$$

[5]

$$= -\frac{1}{f_\alpha}\int_0^\infty [(t\alpha' - cosh(t))^2 cosh(\alpha t) + 2t\alpha' cosh(t)e^{-\alpha t}]e^{-x cosh(t)}dt$$
$$< 0$$

It follows easily from [3] that $\alpha(x) \to \infty$ as $x \to \infty$.

The asymptotic form of $K_\alpha(x)$ for large order and argument is (see ref. 3)

[6] $$K_\alpha(x) \sim \sqrt{\frac{\pi}{2\alpha}}\frac{e^{-\alpha\eta}}{(1+\frac{x^2}{\alpha^2})^{\frac{1}{4}}}$$

with

$$\eta = \sqrt{1+\frac{x^2}{\alpha^2}} - arcsinh\left(\frac{\alpha}{x}\right)$$

and

$$arcsinh(z) = log(z + \sqrt{z^2+1})$$

Substituting expression [6] for $K_\alpha(x)$ in [2] and taking logarithms gives

[7] $$-\eta \sim \frac{1}{2\alpha}log\left(\frac{2\alpha}{\pi}\right) + \frac{1}{4\alpha}log\left(1+\frac{x^2}{\alpha^2}\right) + \frac{logM}{\alpha}$$

For $M = 16^{63}$ and $x > 1$, [7] gives α accurate to three decimal places. $\alpha(x)$ for this value of M is shown in Fig. 1.

If we assume x/α is unbounded as $x \to \infty$, then the left-hand side of [7] is negative and unbounded while the right-hand side is positive as $x \to \infty$. Therefore, assuming $x/\alpha \to a$ as $x \to \infty$, we get the equation

$$\sqrt{1+a^2} + log\frac{a}{1+\sqrt{1+a^2}} = 0$$

which gives $a = 0.662743$. The linear spline which agrees with $\alpha(x)$ at the knots x_i, $0 \le i \le n$, and which has slope $c = 1/a = 1.50888$ for $x > x_n$ is a lower bound to $\alpha(x)$.

3. THE CURVE $e^x K_\beta(x) = M$

The curve $\beta(x)$ defined by the equation

$$e^x K_\beta(x) = M, \quad M > 0 \tag{8}$$

also is a concave function of x with $\beta(x) \to \infty$ as $x \to \infty$. This follows easily by substituting $cosh(t) - 1$ for $cosh(t)$ and β for α in [3], [4] and [5]. A lower bound linear spline can be constructed in a manner similar to the previous case.

$\beta(x)$ is an increasing function of $\sqrt{x}$ but is not a concave function of $\sqrt{x}$. The curves computed by the author each has one point of inflexion so that, for a given M, there is a point x_M for which β is a concave function of $\sqrt{x}$ for $x < x_M$ and a convex function of $\sqrt{x}$ for $x > x_M$. For $M = 16^{63}$, this proposition was checked at the points $\beta = \frac{3}{2}, \frac{5}{2}, ..., 600.5...$ with $\beta(984.7) = 600.5...$ and $\sqrt{x_M} = 4.699...$ This curve is shown in Fig. 2.

Again, using the asymptotic expansion [6] for $e^x K_\beta(x)$ we find from [8] that

$$\frac{x}{\beta} - \eta \sim \frac{1}{2\beta} log\left(\frac{2\beta}{\pi}\right) + \frac{1}{4\beta} log\left(1 + \frac{x^2}{\beta^2}\right) + \frac{logM}{\beta} \tag{9}$$

with

$$\eta = \sqrt{1 + \frac{x^2}{\beta^2}} - arcsinh\left(\frac{\beta}{x}\right)$$

The derivative of $x/\beta - \eta$ with respect to x/β is $1 - \sqrt{1 + \beta^2/x^2}$ which is negative. Hence, $x/\beta - \eta$ decreases monotonically to zero as $x/\beta \to \infty$. If we assume that x/β is bounded for $x \to \infty$, then the right-hand side of [9] goes to zero while the left-hand side is greater than zero as $x \to \infty$. It follows that x/β is unbounded as $x \to \infty$. Expanding [9] in low order terms of β/x gives

$$\beta^2 \sim x(1 + \frac{1}{2x}) log\left(\frac{2xM^2}{\pi}\right) \tag{10}$$

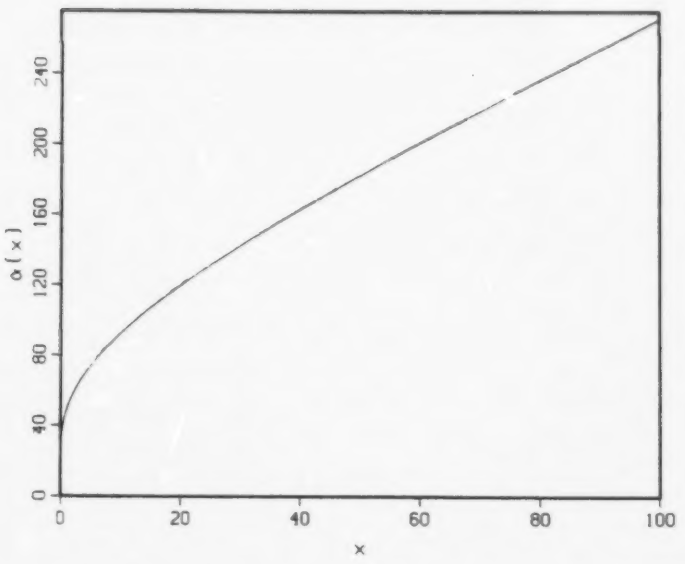

Figure 1. The curve $\alpha(x)$ for $M = 16^{63}$.

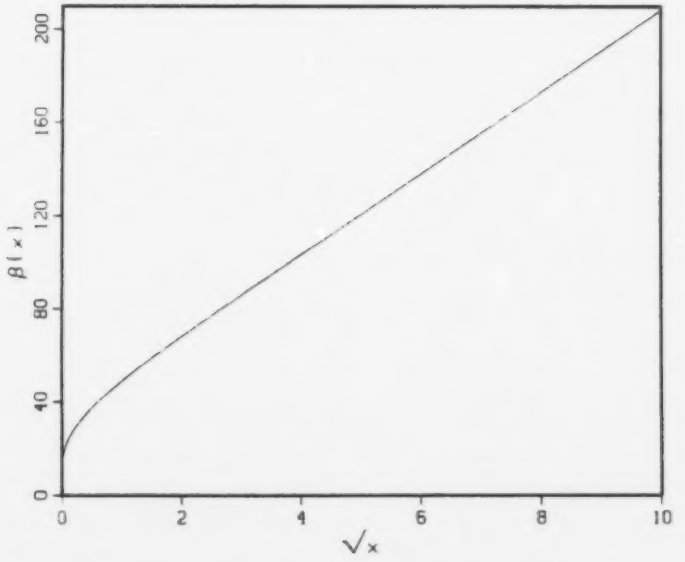

Figure 2. The curve $\beta(x)$ for $M = 16^{63}$.

For $M = 16^{63}$, $\beta(4.283) = 69.5$, and $\beta(984.7) = 600.5$, [10] gives the corresponding values $\beta(4.283) \sim 40.9$ and $\beta(984.7) \sim 592.0$.

An approximation, $a\sqrt{x} + b$, to $\beta(x)$ for $M = 16^{63}$ was calculated with a $17.701 \approx d\beta/d\sqrt{x}\Big|_{\beta=201}$, $b = 31.2233$ and $a\sqrt{x} + b = 201$ when $\beta(x) = 201$. $\beta = 201$ was chosen so that $a\sqrt{x} + b$ intercept $\beta(x)$ for some $x \leq 1$ and is thus, hopefully, a lower bound to $\beta(x)$ for $x > 1$. The error, $E(x) = \beta(x) - a\sqrt{x} - b$, was checked at all values of x in the interval $[1, 984.7]$ for which β is half an odd integer. It was found that $0 \leq E(x) \leq 1.77$ for $1 \leq x \leq 256$ and $E(984.7) = 13.8$.

4. NUMERICAL RESULTS

A subroutine, *SPHERK*, was written in *FORTRAN IV* for the determination of $e^x K_{i+1/2}(x)$, $i = m, m+1, ..., m+n$, for positive values of x. Starting values for forward recurrence were

$$e^x K_{-\frac{1}{2}}(x) = e^x K_{\frac{1}{2}}(x) = \sqrt{\frac{\pi}{2x}}$$

For the computer used, $M = 16^{63} \approx 0.7237 \times 10^{76}$. A lower bound to $\beta(x)$ was calculated as a linear spline with knots x_i, $i = 0, 1, 2, 3, 4$, and elements $a_i x + b_i$ for $x_{i-1} < x \leq x_i$. Table 1 gives this spline and its error in determining $\beta(x)$. *SPHERK* uses as a lower bound to $\beta(x)$ the spline of Table 1 for $x \leq 1$ and the function $a\sqrt{x} + b$ for $x > 1$. Tests for overflow are made in *SPHERK* only for orders greater than this lower bound.

Table 1. Linear spline approximation to $\beta(x)$ for $M = 16^{63}$. On the interval $(x_{i-1}, x_i]$, the spline is equal to $a_i x + b_i$ and the maximum error is determined on that interval.

i	x_i	a_i	b_i	$max(\beta(x) - a_i x - b_i)$
0	0.31×10^{-50}			
1	0.00001	1127586.0	1.499999	8.6
2	0.0625	275.804	12.7731	8.0
3	0.25	39.5817	27.5370	0.88
4	1.0	15.3456	33.5960	1.26
5	4.0	6.43066	42.5110	1.92

A second version, *SPHERL*, of *SPHERK* was written which tests each step of the recurrence formula for overflow. The sequence $e^x K_{i+1/2}(x), \quad i = 0, 1, ..., n,$ for $n + 1/2 < \beta(x)$ was calculated in approximately $5.7n + 28$ μs by *SPHERK* and by *SPHERL* in approximately $8.8n + 24$ μs. The lower bound to $\beta(x)$ used by *RBESK* (1,2) is not as accurate as the one used by *SPHERK*. At points where the accuracies are equal, the above sequence was calculated by *RBESK* in approximately $5.7n + t_0$ μs. t_0 is a function of x with a maximum of approximately 325 at $x = 1$. *RBESK* calculates a sequence of functions $K_{\nu+i}(x), \quad i = 0, 1, ..., n,$ of real orders and uses three different algorithms for the determination of starting values for forward recurrence. These results clearly indicate the increase in efficiency that can be obtained by using a lower bound approximation to $\beta(x)$.

REFERENCES

1. Campbell, J. B. On Temme's algorithm for the modified Bessel function of the third kind. ACM Trans. Math. Software, 6; 581-586; 1980.
2. Campbell, J. B. A FORTRAN IV subroutine for the modified Bessel function of the third kind of real order and real argument. NRC/ERB-925, National Research Council of Canada. 1980.
3. Abramowitz, M. and Stegun, I.A. Handbook of Mathematical Functions. NBS Appl. Math. Series 55, 1964.

Subroutine SPHERK

```
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
C
C.. SUBROUTINE SPHERK(M,N,X,BK,IER)
C
C.. PURPOSE ..
C   COMPUTES THE BESSEL FUNCTION 'K' OF REAL ORDER
C   M+1/2, M+3/2, . . . ., M+1/2+N AND REAL ARGUMENT 'X'.
C
C.. ACCURACY ..
C   DOUBLE PRECISION ACCURACY
C
C.. NOTATION ..
C   THE PRODUCT OF THE BESSEL FUNCTION 'K' OF ORDER 'DMU' AND
C   ARGUMENT 'X' AND DEXP(X) IS DENOTED BY 'K(DMU,X)'.
C   THE MAXIMUM POSITIVE NUMBER ALLOWED BY THE COMPUTER IS DENOTED
C   BY 'DMAX'.
C
C.. DESCRIPTION OF INPUT VALUES ..
C   M, N, X  -  'M' AND 'N' MUST BE GREATER THAN OR EQUAL TO ZERO
C   AND 'X' MUST BE GREATER THAN ZERO. 'X' IS A DOUBLE PRECISION
C   ARGUMENT AND 'M' AND 'N' ARE INTEGER ARGUMENTS.
C
C.. DESCRIPTION OF OUTPUT VALUES ..
C   BK  -  DOUBLE PRECISION ARRAY OF DIMENSION AT LEAST N+1
C   WHICH CONTAINS THE COMPUTED VALUES OF THE BESSEL FUNCTIONS.
C   K(M+1/2+I,X) IS RETURNED IN BK(I+1) IF IT IS LESS THAN 'DMAX'.
C   OTHERWISE, K(M+1/2+I,X)/K(M-1/2+I,X) IS RETURNED.
C   IER  -  'IER' CAN HAVE THE VALUES -2, -1, 1, 2, . . ., N+1.
C   FOR IER < 0, INCORRECT VALUES ARE RETURNED IN THE ARRAY 'BK'
C   AND FOR IER > 0, CORRECT VALUES ARE RETURNED. THE FOLLOWING
C   CONDITIONS ARE DEFINED BY 'IER':
C   (1) FOR INVALID ARGUMENTS M < 0, N < 0 OR X.LE.0, IER=-2.
C   (2) IF K(M+1/2,X) OR K(M+1/2+N,X)/K(M-1/2+N,X) IS GREATER
C   THAN OR EQUAL TO 'DMAX', IER = -1.
C   (3) IF K(M+1/2+N,X) IS LESS THAN 'DMAX', IER = N+1.
C   (4) IF K(M+1/2+N,X) IS GREATER THAN OR EQUAL TO 'DMAX' AND
C   K(M+1/2+N,X)/K(M-1/2+N,X) IS LESS THAN 'DMAX', 'IER' IS THE
C   SMALLEST VALUE OF THE INTEGER 'I' FOR WHICH K(M+1/2+I,X) IS
C   GREATER THAN OR EQUAL TO 'DMAX'.
C
C.. FORTRAN SUPPLIED MATHEMATICAL SUBPROGRAMS CALLED ..
C   DFLOAT, DSQRT
C
C . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
      SUBROUTINE SPHERK(M,N,X,BK,IER)
      DOUBLE PRECISION BK(1),BK1,BK2,C,D,DFLOAT,DMAX,DSQRT,THONU,X
      REAL A(5),B(5),XXNOT(4)
      DATA D/Z411400931FF62706/, DMAX/Z7FFFFFFFFFFFFFFF/,
     $     XXNOT/0.00001,   0.0625,   0.25,     1.0/,
     $        A/1127586.,  275.804,  39.5817,  15.3456,  17.7010/,
     $        B/0.999999,  12.2731,  27.0370,  33.0960,  30.7233/
C.. D = DSQRT(PI/2.D0)
C.. DMAX = 16**63.
C
      EX=X
      IER=-2
      IF(M.LT.0.OR.N.LT.0.OR.EX.LE.0.0) GO TO 80
C
C.. CALCULATION OF K(-1/2,X) AND K(1/2,X)
```

```
      NPLUS1=N+1
      C=DSQRT(X)
      BK1=0/C
      BK(1)=BK1
      IEND=N+M
      IER=NPLUS1
      IF(IEND.EQ.0) GO TO 80
      THONU=-1.D0
      BK2=BK1
      J=1-M
C
C.. CALCULATION OF A LOWER BOUND 'MINF' TO THE VALUE OF 'L' FOR
C.. WHICH K(L+1/2,X) = DMAX.
      IF(EX.GT.XKNOT(1)) GO TO 10
      MINF=A(1)*EX+B(1)
      IF(MINF.GT.0) GO TO 20
      IF(2.D0*DFLOAT(M+N)-1.D0.GT.X*DMAX) GO TO 60
      GO TO 40
   10 I=2
      IF(EX.GT.XKNOT(3)) I=4
      IF(EX.GT.XKNOT(I)) I=I+1
      IF(I.EQ.5) EX=C
      MINF=A(I)*EX+B(I)
C
C.. CALCULATION OF 'IER', K(M+1/2+I,X), I = 0, 1, . . ., IER-1,
C.. K(M+1/2+I,X)/K(M-1/2+I,X), I = IER, IER+1, . . ., N
   20 IF(MINF.GT.IEND) MINF=IEND
      DO 30 I=1,MINF
        C=BK1
        BK1=BK2
        THONU=THONU+2.D0
        BK2=THONU/X*BK1+C
        J=J+1
        IF(J.GT.0) BK(J)=BK2
   30 CONTINUE
      IF(MINF.EQ.IEND) GO TO 80
   40 C=BK2/BK1
      MPLUS1=MINF+1
      DO 50 I=MPLUS1,IEND
        THONU=THONU+2.D0
        C=THONU/X+1.D0/C
        J=J+1
        IF(BK2.GE.DMAX/C) GO TO 60
        BK2=C*BK2
        IF(J.GT.0) BK(J)=BK2
   50 CONTINUE
      GO TO 80
   60 IER=-1
      IF(J.LE.1) GO TO 80
      IER=J-1
      BK(J)=C
      J=J+1
      DO 70 I=J,NPLUS1
        THONU=THONU+2.D0
        C=THONU/X+1.D0/C
        BK(I)=C
   70 CONTINUE
   80 RETURN
      END
```